

# The Community Hydrologic Prediction System: a Services Oriented Architecture - OHD Experiences

Pedro Restrepo

Jon Roe

Office of Hydrologic Development

6 December, 2005

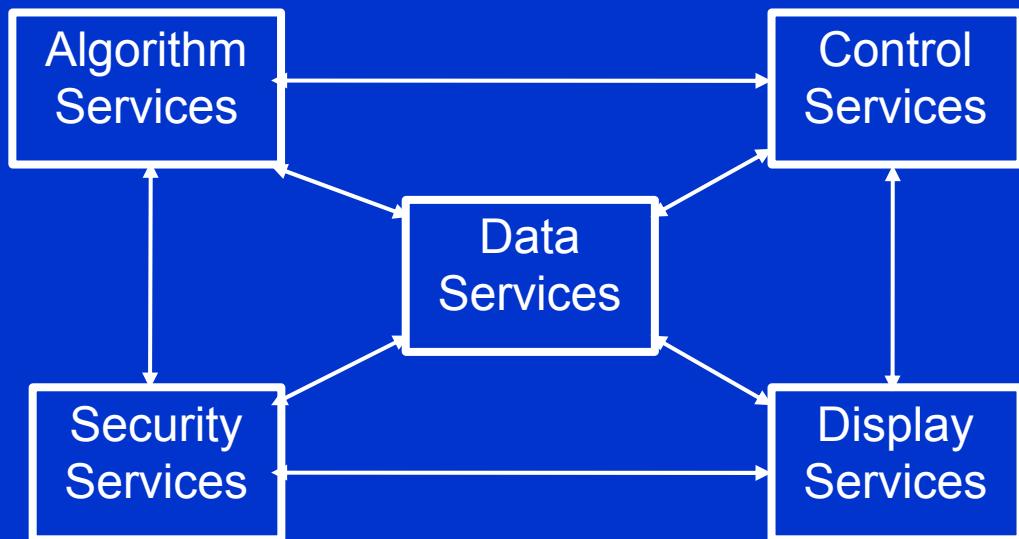
# Contents

- Why CHPS?
- Service-Oriented Architecture
- Review of accomplished work
- Concerns and Experiences

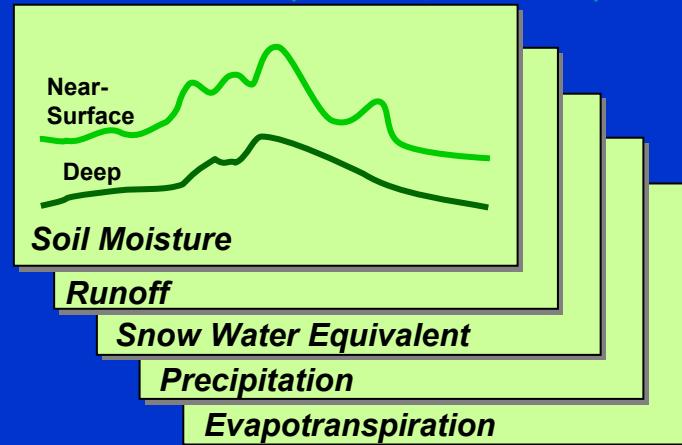
# CHPS – Why?

- Infuse new science into NWS operations
- Provide access to an expanded set of hydrometeorologic, hydrologic, and hydraulic models
- Enable fine space and time scale distributed hydrologic modeling
- Introduce parallel processing for ensemble predictions
- Support concurrent, distributed development
- Encourage scientific collaboration
- FLEXIBILITY

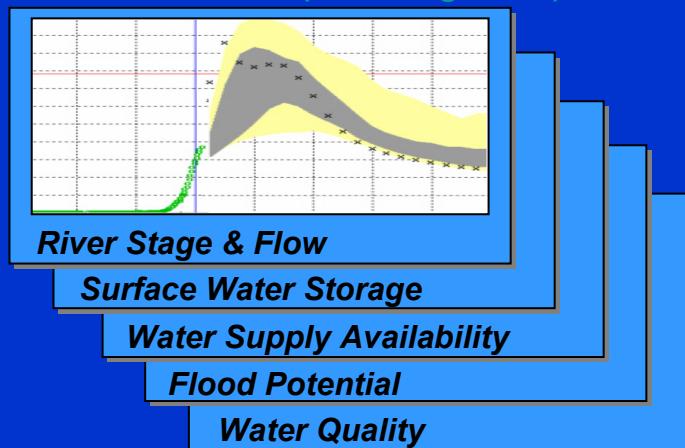
# Community Hydrologic Prediction System



Point Information (Stations, Grid Cells)



Vector Information (River Segments)



Hydrology XML Consortium

<http://www.weather.gov/oh/hydroxc/index.html>

# Review of OHD's work on CHPS

*3 years of work!*

# Case for CHPS: NOAA

In the **NOAA 2007 Annual Guidance Memo (AGM)** VADM Lautenbacher has identified several key priorities:

- (p.3) “We must employ a **consistent data management strategy** that facilitates data discovery and access for our users”
- (p.4) “We must....**improve existing regional coordination** across NOAA and with other federal agencies
- (p.5) “There is a **paramount need** to provide a transition from research to **products and services** that provide useful information for national and regional management decisions”
- (p.5) “We must provide **national leadership**, foster innovative partnerships, and support collaborative research”
- (p.6) “**Integrated information services delivery** is a long-term NOAA objective”

# Case for CHPS: NWS

## From NWS Strategic Plan for 2005-2010: Working Together to Save Lives:

“To meet these expanding requirements, our weather, space weather, air quality and water predictions and the information we disseminate need to be at the limits of the skill which science, technology, and a highly-trained workforce can provide. We are committed to expand these limits by enhancing observing capabilities; by **improving data assimilation** to use effectively all the relevant data we and others collect; by **improving collaboration with the research community** through creative approaches like community modeling (e.g., establish an Earth System Model Framework); by **quickly transforming scientific advances** in modeling into improved operational products; by improving the techniques used by our expert forecasters; by evolving our services from a text-based paradigm to one based on making NWS and NOAA information available quickly, efficiently, and in convenient and understandable forms (e.g., National Digital Forecast Database and GIS); by including information on forecast uncertainty to enhance customer decision processes; by **taking advantage of existing and emerging technologies to disseminate this information**; by expanding our outreach and education efforts to better meet the needs of a more diverse population; and by maintaining an up-to-date technology base and a workforce trained to use all of these tools to maximum effect.”

# Case for CHPS: OHD needs

## OHD needs:

- Support more scientific and community collaboration
- Streamline research-to-operations process
- Improve efficiency of operational system resources (e.g., concurrent processing)
- Improve site flexibility to run additional models
- Support new data formats as they emerge (e.g., grids)
- Modernization
  - flexibility to evolve
  - improve software maintainability

## Constraints:

- Cannot afford to discard large software investment
- RFCs must continue to get their jobs done during transition

# What is SOA?

- Service-oriented architecture is an **emerging standard** for large scale design of software systems.
- A service is defined as: “Any act or performance that one party can offer to another that is essentially intangible and does not result in the ownership of anything. Its production may or may not be tied to a physical product.”
- Thinking of software as service encourages us to think about what the output is, not how the output is created.
- Software as service emphasizes the interchangeability of how the service is rendered.
- Services thus emphasize 2 key points:
  - **loose coupling**
  - **interchangeability of service implementation.**

# Why Service-oriented Architecture?

What can you do with service-oriented architecture?

- Integrate applications written in different languages, running on different platforms, working with different databases
- Access functionality provided by other organizations
- Integrate functionality across organizations
- Share computing resources in central locations
- Access the resources of distributed computing systems
- Reduce dependencies on service implementations
- Flexibly interchange service implementations

# Concerns with SOA

SOA raises concerns for building applications that are not present in traditional applications. Application designs will need to consider these issues:

- Availability
  - While services may typically be reliable, they may sometimes not be available.
- Performance
  - Distributed computing scales well, however if XML processing and network messaging can slow down applications, if used inappropriately.
- Security
  - By making functionality available over networks, security must be addressed.
- Cost of complexity
  - Service-oriented architecture adds new components to a system, such as messaging protocols, networking issues, servers and infrastructure requirements. This complexity has costs. SOA should be applied when the benefits outweigh these costs.

# OHD Experiences with SOA

- Summer 2002 – OHD engages Apex to build modern messaging/logging API for NWSRFS as a Proof-of-Concept
  - gauge possible longer-term association
- Fall 2002 – Apex presents new architecture vision
  - understanding of OHD needs demonstrated
  - more POC work identified
- Summer 2003 – Apex builds workflow management Proof-of-concept using SOA techniques
  - J2EE application server (using BEA WebLogic), Java message service, logging server (using Oracle RDBMS)
  - configuration and service messaging using XML
  - active application management, encapsulated legacy Fortran
  - cross-platform distributed computing (Linux/Windows)

# OHD Experiences with SOA (cont'd)

- Winter 2004 – Apex presents full SOA model (5 services)
- Spring 2004 – start four-phase river, reservoir, snow (RRS) Pre-processor project
  - flesh out SOA Data Services components
  - show SOA can work in real world environment
- Summer 2004 – OHD pursues Apex recommendation to explore standardization of data messaging for SOA
  - Hydrology XML Consortium (HydroXC) formed 29 September 2004
- Winter 2005 – demos of working Data Services for RRS
  - “FS5file” access with and without Data Services
  - PostgreSQL IHFS\_DB access
  - complete RRS Pre-processor Data Services POC

# OHD Experiences with SOA (cont'd)

- Spring 2005 – Apex concludes version 1 of HydroXC data messaging schema
- Summer 2005 – design elements to move RRS Pre-processor POC into operations
- Summer 2005 – start several other specific projects
  - MAT Pre-processor architecture shell (following RRS)
  - Use Data Services concepts for Distributed Hydrologic Modeling (DHM) implementation
  - Control Services via NWSRFS ESP (consider DELFT-FEWS)
  - Extend/refine HydroXC schema with member and RFC workshops
- Summer 2005 – acquire onsite SOA specialist (former Apex co-developer of POCs)

# OHD Experiences with SOA (cont'd)

September 2005 -- met with WL | Delft Hydraulics:

- WL | Delft Hydraulics has developed a system called DELFT-FEWS that relies on a J2EE-based architecture to provide data, workflow, and grid computation services.
- Some features of DELFT-FEWS
  - Provide central data services that store all observations and operational forecasts. Local clients communicate with the central data services via Java Message Service (JMS).
  - Central services store workflows of techniques and models used operationally.
  - Central services allow workflows to be executed on a cluster of servers.
- Their system is operational in the U.K. and other countries.

# OHD CHPS Future Plans

- Continue to design and develop architecture
  - Applicability of DELFT-FEWS to OHD?
- Continue prototyping Data, Control, & Algorithm Services on OHD projects with Apex and RTi:
  - Keep existing software where necessary (cost)
  - Upgrade key existing software components to support SOA
  - Design new software to be consistent with SOA
- Operationally field hydro applications that start to use SOA
  - RRS & MAT Pre-processors (Data and Algorithm Services)
  - DHM (Data Services)
  - NWSRFS ESP (Control Services)
  - USACE HEC ResSim & HEC-RAS (Algorithm Services)

# OHD CHPS Future Plans (cont'd)

- Evaluate progress toward target architecture, adjust priorities as needed
- Work with NWS International Activities Office to use hydro projects for other countries to help move elements of CHPS forward where appropriate
  - Mekong River project for Vietnam
  - Flood Early Warning Project for Romania
  - Consolidate hydro internal & international capabilities

# Thank you